

Optimal Forwarding in Opportunistic Delay Tolerant Networks with Meeting Rate Estimations

Shohreh Shaghaghian and Mark Coates

Abstract

Data transfer in opportunistic Delay Tolerant Networks (DTNs) must rely on unscheduled sporadic meetings between nodes. The main challenge in these networks is to develop a mechanism based on which nodes can learn to make nearly optimal forwarding decision rules despite having no a-priori knowledge of the network topology. The forwarding mechanism should ideally result in a high delivery probability, low average latency and efficient usage of the network resources. In this paper, we propose both centralized and decentralized single-copy message forwarding algorithms that, under relatively strong assumptions about the networks behaviour, minimize the expected latencies from any node in the network to a particular destination. After proving the optimality of our proposed algorithms, we develop a decentralized algorithm that involves a recursive maximum likelihood procedure to estimate the meeting rates. We confirm the improvement that our proposed algorithms make in the system performance through numerical simulations on datasets from synthetic and real-world opportunistic networks.

Index Terms

Delay Tolerant Networks, Opportunistic Forwarding, Meeting Rate Estimation

I. INTRODUCTION

DELAY or *Disruption Tolerant Networks* (DTNs) are a class of wireless mobile node networks in which the communication path between any pair of nodes is frequently unavailable. Nodes are thus only intermittently connected. DTNs were first studied in the 1990s when the research community considered how the Internet could be adapted for space communications [1]. Later, it was recognized that DTNs were a suitable model for several terrestrial networks.

The authors are with the Computer Networks Research Laboratory, Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada (e-mail: shohreh.shaghaghian@mail.mcgill.ca; mark.coates@mcgill.ca). This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) via grant application 261528.

Manuscript received April 3, 2015

DTNs can be categorized according to whether the node connections are scheduled (thus predictable) or random (hence unpredictable). Space communication networks fall into the first group. Networks belonging to the second category, which are the focus of this paper, are also referred to as *opportunistic networks*, because nodes seize the opportunity to transfer data when a communication channel becomes available. Opportunistic networks have been studied intensively in recent years (e.g., [2]–[4]) because they can fulfil a number of useful purposes, such as non-intrusive wildlife tracking (e.g., ZebraNet [5] and SWIM [6]), emergency response in disaster scenarios (e.g., ChaosFIRE [7]), provision of data communication to remote and rural areas (e.g., DakNet [8]), and traffic offloading in cellular networks (e.g., [9]).

In most opportunistic networks, the nodes are highly mobile and have a short radio range, and the density of nodes is low. In many cases, nodes have limited power and memory resources. These attributes combine with the intermittent connections to make routing traffic challenging. Routing is usually based on a store-carry-forward mechanism that exploits node mobility. In this mechanism, the source transmits its message to a node it meets. This intermediate node stores and then carries the received message until it meets another node to which it can forward the message. This process is repeated until the message reaches its destination. The key ingredients in designing an opportunistic network routing protocol are the forwarding decisions: should a node forward a message to a neighbour it meets? should it retain a copy for itself?

Although much research effort has been devoted to the development of opportunistic network routing algorithms [10]–[27], the algorithms are either centralized, have no performance guarantees, or ignore the need to estimate network parameters. Our work focuses on the mobile ad-hoc network (MANET) setting, where node speed is much reduced compared to the vehicular ad-hoc (VANET) case, and we can assume that there are fewer restrictions on the amount of data that nodes can transfer when they meet. In this paper, we derive a decentralized routing algorithm that has performance guarantees (under simplifying assumptions about the network behaviour). When the meeting times between nodes are independent and exponentially distributed, the routing algorithm minimizes the expected latency in sending a packet from any source node to a specific destination. We examine the behaviour of the routing algorithm when the meeting rates are learned online using a recursive maximum likelihood procedure. We show that, for a stationary network, the decision rules and achieved expected latencies converge to those obtained when there is exact knowledge of the meeting rates. We present the results of simulations that compare the performance of the proposed algorithm to previous approaches, and examine how the algorithm is affected by practical network limitations (finite buffers, restrictions on data exchange, message expiry times).

Organization: The paper is organized as follows. In the following subsection, we discuss related work. In Section II, we describe our system model and formulate the routing problem. In Section III, we present the forwarding algorithms and discuss their optimality under the network modeling assumptions. We present numerical

simulation results in Section IV and make concluding remarks in Section V.

A. Related Work

The first proposed approaches for routing in opportunistic networks were based on extending the concept of flooding to intermittently connected mobile networks. In these *replication-based* methods, a node forwards the messages stored in its buffer to all of (or to a fraction of) the nodes it encounters. There is no attempt to evaluate the capability of a given node to expedite the delivery. These routing algorithms have few parameters: they determine only how much replication can occur and which nodes can make copies of packets. One of the earliest algorithms was *epidemic routing* [10], in which a node forwards a message to any node it meets, provided that node has not previously received a copy of the message. Thus messages are quickly distributed through the connected portions of the network. Other replication-based approaches ([11], [12], [28], [29]) manage to reduce the transmission overhead of epidemic routing and improve its delivery performance through modification of the replication process and prioritization of messages. Models have been developed that allow an analytical characterization of the performance of the epidemic routing techniques [30], [31]. Replication-based approaches result in a high probability of message delivery since more nodes have a copy of each message, but they can produce network congestion.

A step towards achieving more efficient routing approaches is to consider the history of node contacts in the network instead of blindly forwarding packets. *History-based* (also called *utility-based*) routing algorithms assume that nodes' movement patterns are not completely random and that future contacts depend on the frequency and duration of past encounters. Based on these past observations, both the source and the intermediate nodes decide whether to forward a message to nodes they encounter or to store it and wait for a better opportunity. An early example is [13], which extends epidemic routing to situations with limited resources, incorporating a dropping strategy for the case when the buffer of a node is full. The dropping decisions are based on the meeting history of the node. PROPHET [14] assigns a delivery probability metric to each node which indicates how likely it is that the message will be delivered to the destination by that particular node. This metric is updated each time two nodes meet, and thus takes into account the history of meetings in the network. MaxProp [15] and MEED [16] are other examples of history-based algorithms proposed for vehicular DTNs. In these networks, nodes move with higher speeds, reducing the amount of time they are in each other's radio range. Hence, the two main limiting resources are the duration of time that nodes are able to transfer data and their storage capacities.

Other researchers have examined whether it is possible to exploit other characteristics of opportunistic networks to improve the performance of routing algorithms. Since social interactions often determine when connections between nodes occur, several algorithms strive to use social network concepts like betweenness centralities (e.g. [17]), or community formations (e.g. [18]–[20]). Other algorithms have attempted to take advantage of the strategic

behaviour of nodes (e.g. [21], [22]). Our work focuses on routing a message to a single destination, but there are connections to research that addresses the task of spreading information to multiple nodes in a network. Of particular interest is the gossip-based approach in [23], which greatly reduces the number of message copies in the network while achieving near-optimal dissemination.

The experimental-based studies demonstrate the efficiency of their proposed methods by running simulations on traces recorded from real world opportunistic networks. Experimental analyses are valuable and take into account practical considerations, but they can leave us with an incomplete understanding of how an algorithm operates and how it will perform in other untested network conditions. For example, the behaviour of PROPHET has been shown to be very sensitive to parameter choice [32]. It is also useful to design an optimal algorithm under slightly less realistic modeling assumptions, and then consider how it can be adapted to address the practical limitations, without completely losing its desirable features. More recent studies have focused on deriving a forwarding process whose optimality (in some sense) can be mathematically proved under assumptions about network behaviour. [24] extends the two hop relay strategy of [11] by considering the expected delivery time to the destination as a metric to find the best set of candidate relays. By increasing the number of relaying steps recursively, a centralized single-copy multi-hop opportunistic routing scheme is proposed for sparse DTNs. The main defect of a centralized approach is that global knowledge of the network is required in order to make forwarding decisions.

There have been some efforts towards migrating to decentralized solutions that still provide performance guarantees. [25] proposes a decentralized time-sensitive algorithm called TOUR in which message priority is taken into account in addition to nodes' expected latencies when making forwarding decisions. Although in TOUR each node only needs to be aware of the local information about the rates of contacts with its own set of neighbours, the algorithm assumes that the node knows the exact contact rates. In most practical scenarios, this assumption is not valid.

Some researchers have explored how imprecision in the measurement or estimation of network parameters can impact the performance of opportunistic network routing algorithms. In [26], [27], Boldrini et al. discuss different sources of errors that may exist in parameter estimation like missed encounters, incorrect combination of short contacts, and memory limitations. They model these errors as a random variable with a normal distribution and evaluate the performance of four different forwarding schemes under this model. Although this error analysis is useful, Boldrini et al. do not specify how parameters should be estimated in order to obtain a performance that approaches what can be achieved when perfect a-priori knowledge of the network parameters is available.

Some of the results in this paper were presented in an earlier conference paper [33], but here we include more extensive experimental analysis and additional theoretical results.

II. SYSTEM MODEL

We consider a network of N mobile nodes which aim to send messages to a particular destination node d . The set of nodes is denoted by \mathcal{N} . We assume that the random inter-meeting times of nodes are independent and exponentially distributed with parameter λ_{ij} for nodes i and j .

Although the aggregate intermeeting distributions of nodes in mobile ad-hoc networks often follow a truncated power law [34], [35], there is evidence that the intermeeting times of individual pairs of nodes can be adequately modeled by exponential distributions with heterogeneous coefficients [36]–[39]. In particular, Conan et al. [36] and Gao et al. [37] conduct statistical analyses of mobile social network data traces, including the Infocom data set [40] that we analyze in Section IV. They demonstrate that most pairs of nodes have intermeeting times that are approximately exponentially distributed. In [38], [39], approximately exponential distributions of individual meeting times are detected through statistical analyses of car/taxi mobility traces.

We associate with the network a *contact graph* which is formed by adding a link between any two nodes that meet. We assume that the contact graph is connected and denote the set of neighbors of node i in this graph by \mathcal{S}_i . Since the contacts between nodes are not pre-scheduled, we cannot identify end-to-end paths ahead of time. Hence, solving the routing task is equivalent to identifying the forwarding decisions that nodes should make when meeting each other. We assume that nodes' buffer sizes are unlimited, message Time To Live (TTL) is infinity and that nodes' speed and message lengths are such that any number of messages can be forwarded during each meeting. We consider only algorithms that do not involve replication. In the class of algorithms we consider, each time node i meets one of its neighbors $j \in \mathcal{S}_i$, it forwards a message destined for d with probability p_{ij} . Considering the matrix $\mathbf{P}_{N \times N}$ comprised of all pairs i and j , we set $p_{ij} = 0$ if nodes i and j never meet and are thus not neighbors in the contact graph. We denote the forwarding probabilities of node i by the vector \mathbf{p}_i ; this is the i -th row of the matrix \mathbf{P} .

The expected latency from node i to destination d is a function of the probability decision matrix \mathbf{P} and we denote it by $L_{id}(\mathbf{P})$. Our goal is to find the matrix \mathbf{P}^* such that the sum of the expected latencies of all the nodes in the network to the specified destination d is minimized. Let us call this utility function $\mathcal{U}(\mathbf{P}) = \sum_{i \in \mathcal{N}} L_{id}(\mathbf{P})$. We assume that the network topologies and meeting rates are such that the solution \mathbf{P}^* is unique. If not, our algorithms guarantee that we reach one of the optimal matrices, but the proofs are more complicated. The first step towards achieving this goal and finding matrix \mathbf{P}^* is to discover how the expected latency of an arbitrary node i , $L_{id}(\mathbf{P})$, depends on the elements of the probability decision matrix \mathbf{P} in general. Lemma 1 provides an expression for $L_{id}(\mathbf{P})$ in terms of \mathbf{P} and $\lambda_{ij}, j \in \mathcal{S}_i$. The proof is available in Appendix A.

Lemma 1. *The expected latency of a node $i \in \mathcal{N}$ to the destination d is*

$$L_{id}(\mathbf{P}) = \frac{1 + \sum_{j \in \mathcal{S}_i} p_{ij} \lambda_{ij} L_{jd}(\mathbf{P})}{\sum_{j \in \mathcal{S}_i} p_{ij} \lambda_{ij}} \quad (1)$$

Based on the expression derived in Lemma 1, the expected latency of each node to the destination depends on the expected latencies of its neighbours. This result raises a substantial question: Does the probability decision matrix that minimizes the sum of expected latencies of all nodes of the network (\mathbf{P}^*), also minimize the expected latency of each individual node? Before continuing to propose algorithms for finding \mathbf{P}^* , we answer this question and make two points about the structure of \mathbf{P}^* through the following theorem. The proof is provided in Appendix B.

Theorem 1. *Suppose $\mathbf{P}^* = \arg \min_{\mathbf{P} \in [0,1]^{N \times N}} \sum_{i=1}^N L_{id}(\mathbf{P})$. Then:*

- (1) \mathbf{P}^* is a binary matrix (its components are either 0 or 1).
- (2) For any $i \in \mathcal{N}$, the matrix \mathbf{P}^* also minimizes $L_{id}(\mathbf{P})$:

$$\forall i \in \mathcal{N} : \quad \mathbf{P}^* = \arg \min_{\mathbf{P} \in [0,1]^{N \times N}} L_{id}(\mathbf{P}) \quad (2)$$

Theorem 1 shows that the minimization problem is actually a binary problem. Each time node i meets one of its neighbours $j \in \mathcal{S}_i$, it either forwards the message or keeps it. From now on, we change our notation and use the binary indicator matrix \mathbf{B} instead of \mathbf{P} to capture this binary decision. Therefore, the optimization takes the form:

$$\mathbf{B}^* = \arg \min_{\mathbf{B} \in \{0,1\}^{N \times N}} \sum_{i=1}^N L_{id}(\mathbf{B}) \quad (3)$$

Theorem 1 also states that the optimum solution matrix \mathbf{B}^* can be equivalently achieved by minimizing the expected latency of each of the network nodes to the destination. This is the main idea of developing centralized and decentralized algorithms for finding \mathbf{B}^* . In the next section, we introduce the algorithms we have proposed for solving this optimization problem and prove that they find the optimal solution.

III. ALGORITHMS

In the first part of this section, we try to find \mathbf{B}^* in a centralized fashion where the whole topology and meeting rates of the network are available at a central unit. This unit calculates a binary matrix \mathbf{B} and informs the nodes about the neighbours they should forward their buffered messages to. We prove that the solution achieved upon completion of this algorithm is the same as the optimum solution \mathbf{B}^* . In the second part of the section, we introduce a decentralized algorithm and prove that it converges to the same global solution with probability 1. The advantage of the decentralized approach is that no node needs to have a global knowledge of the network and each node can learn its own optimal forwarding decisions. The only piece of information a node needs to know is its

meeting rates with its own neighbours. Finally in the last part of the section, we make our model more realistic by assuming that nodes have no a-priori knowledge of any meeting rates. In this more practically realistic scenario, each node estimates the meeting rates with its neighbours, updating its estimates each time a contact occurs.

Algorithm 1 Centralized Greedy Latency Minimization

```

1: // Initialization
2:  $\mathcal{A} = \{d\}$ ,  $\mathbf{B} = \mathbf{0}_{N \times N}$ ,  $L_{dd} = 0$ ,  $L_{jd} = \infty$  for  $j \neq d$ 
3: // Iteratively add nodes to the set
4: while  $\mathcal{A} \neq \mathcal{N}$  do
5:   for each node  $i \notin \mathcal{A}$  do
6:     Identify  $\mathcal{S}_{i\mathcal{A}} = \mathcal{S}_i \cap \mathcal{A}$ 
7:     Calculate  $G_{id}$  for  $\mathbf{m}_i^* \in \{0, 1\}^{|\mathcal{S}_{i\mathcal{A}}|}$ 
           
$$G_{id} = \min_{\mathbf{m}_i} \frac{1 + \sum_{j \in \mathcal{S}_{i\mathcal{A}}} m_{ij} \lambda_{ij} L_{jd}}{\sum_{j \in \mathcal{S}_{i\mathcal{A}}} m_{ij} \lambda_{ij}}$$

8:     Denote  $\mathcal{D}_i = \{j | j \in \mathcal{S}_{i\mathcal{A}}, m_{ij}^* = 1\}$ 
9:   end for
10:  Identify  $v = \arg \min_{i \in \mathcal{N}/\mathcal{A}} G_{id}$ 
11:  Set  $L_{vd} = G_{vd}$ 
12:  Set  $b_{vj} = 1$  for all  $j \in \mathcal{D}_i$ 
13:  Set  $\mathcal{A} = \mathcal{A} \cup \{v\}$ 
14: end while

```

A. Centralized Approach with Global Knowledge

Suppose for each node $i \in \mathcal{N}$, the set of neighbours \mathcal{S}_i and their meeting rates $\lambda_{ij}, j \in \mathcal{S}_i$ are known at a central calculation unit. Algorithm 1 presents an iterative procedure to identify a binary decision matrix \mathbf{B} . In this algorithm, we first decide on the forwarding rules of the node that has the most frequent direct contacts with the destination. We refer to this node as node A . In order to achieve the minimum expected latency to the destination, node A should forward its generated or received messages only to the destination, ignoring its meetings with any other nodes. All other nodes that A encounters meet the destination less frequently and, if they forward their messages to the destination through other nodes, these other nodes also meet the destination less frequently than A . In subsequent steps of the algorithm, we consider all the nodes that have direct contacts with the nodes whose forwarding decision rules have already been made (the set \mathcal{A}) and calculate the minimum latency that each of them can obtain by forwarding through nodes in \mathcal{A} to the destination. At the end of each iteration, we finalize the forwarding decision for the one node that can achieve the minimum latency and add it to \mathcal{A} . We repeat the procedure until the decision has been made for all the nodes of the network and the elements of the binary matrix \mathbf{B} have all been specified. The next theorem states that the binary matrix \mathbf{B} resulting from applying this procedure, as specified concretely in Algorithm 1, achieves the minimum sum of expected latencies to the destination. The proof can be found in Appendix C.

Theorem 2. Suppose all meeting rates are different and there exists a unique solution \mathbf{B}^* for the optimization problem (3).

1) After each iteration of Algorithm 1,

- a) $\forall i \in \mathcal{A}, \forall j \in \mathcal{N} : b_{ij} = b_{ij}^*$
- b) $\forall i \in \mathcal{A} : L_{id}(\mathbf{B}) = L_{id}(\mathbf{B}^*)$
- c) $\max_{i \in \mathcal{A}} L_{id}(\mathbf{B}^*) < \min_{i \notin \mathcal{A}} L_{id}(\mathbf{B}^*)$

2) Upon completion, Algorithm 1 identifies a labelling \mathbf{B} and associated expected latencies L_{id} such that $\mathbf{B} = \mathbf{B}^*$

Theorem 2 demonstrates that the iterative optimization procedure expressed in Algorithm 1 finds the solution of the minimization problem in (3). If there is not a unique solution, then at some point in Algorithm 1, there will be multiple \mathbf{m}_i that solve the optimization in line 7. It is straightforward to show that choosing any one of these \mathbf{m}_i leads to a decision matrix \mathbf{B} that achieves the minimum expected latencies.

B. Decentralized Approach with Partial A-priori Knowledge

Suppose no central unit exists and each node is just aware of its own \mathcal{S}_i and the meeting rates $\lambda_{ij}, j \in \mathcal{S}_i$. Algorithm 2 demonstrates how nodes can make their binary forwarding decisions based on this local information. Since the expected latency of each node depends on the expected latency values of its neighbours, nodes need to have an estimation of their neighbours' expected latencies to be able to make forwarding decisions. We denote by $\hat{L}_{id}(j)$ the estimate at node j of the latency from node i to the destination. In Algorithm 2, each time two nodes meet, they update these estimates and then recalculate their optimum forwarding rules. Theorem 3 proves that this decentralized approach results in the same global optimum solution. The proof of Theorem 3 is provided in Appendix D.

Algorithm 2 Decentralized Greedy Latency Minimization

```

1: // Initialization
2:  $\mathbf{B} = \mathbf{0}_{N \times N}$ 
3:  $\forall i \in \mathcal{N}/d, \forall j \in \mathcal{N} : \hat{L}_{dd}(j) = 0, \hat{L}_{id}(j) = \infty$ 
4: while Nodes continue to meet do
5:   // Nodes  $i$  and  $j$  meet at time  $t$ 
6:   Set  $\hat{L}_{id}(j) = \hat{L}_{id}(i)$ 
7:   Set  $\hat{L}_{jd}(i) = \hat{L}_{jd}(j)$ 
8:   Update  $\hat{L}_{id}(i) = \min_{\mathbf{m}_i \in \{0,1\}^{|\mathcal{S}_i|}} \frac{1 + \sum_{k \in \mathcal{S}_i} m_{ik} \lambda_{ik} \hat{L}_{kd}(i)}{\sum_{k \in \mathcal{S}_i} m_{ik} \lambda_{ik}}$  and identify the minimizing  $\mathbf{m}_i^*$ 
9:   Update  $\hat{L}_{jd}(j) = \min_{\mathbf{m}_j \in \{0,1\}^{|\mathcal{S}_j|}} \frac{1 + \sum_{k \in \mathcal{S}_j} m_{jk} \lambda_{jk} \hat{L}_{kd}(j)}{\sum_{k \in \mathcal{S}_j} m_{jk} \lambda_{jk}}$  and identify the minimizing  $\mathbf{m}_j^*$ 
10:  Set  $\mathbf{b}_i = \mathbf{m}_i^*$  and  $\mathbf{b}_j = \mathbf{m}_j^*$ 
11: end while

```

Theorem 3. *The decision matrix \mathbf{B} identified by Algorithm 2 converges to \mathbf{B}^* with probability 1.*

We refer to our proposed decentralized greedy latency minimization algorithm (Algorithm 2) as MinLat and evaluate its efficiency in different random and real-world networks based on certain performance metrics in Section IV. Regarding the computational complexity of finding the minimum expected latency in MinLat, the following lemma shows that the optimizations in lines 8 and 9 of this algorithm are linear fractional programs and can be solved quickly using variants from linear programming. Further details are available in Appendix E.

Lemma 2. *The minimization problem in Algorithm 2,*

$$\hat{L}_{id}(i) = \min_{\mathbf{m}_i \in \{0,1\}^{|S_i|}} \frac{1 + \sum_{k \in S_i} m_{ik} \lambda_{ik} \hat{L}_{kd}(i)}{\sum_{k \in S_i} m_{ik} \lambda_{ik}}, \quad (4)$$

can be converted to a linear programming problem.

Assuming that (4) can be solved in polynomial order $P(|S_i|)$, the worst case complexity order of Algorithm 1 is $O(N^2)P(N)$ because in the i th round of this algorithm, (4) should be solved for each of the $N - i$ nodes that are not in the set \mathcal{A} . In Algorithm 2, each time node i meets one of its neighbours, it solves a problem of complexity $P(|S_i|)$. The only information that a node needs to share when it meets another node is its estimate of its own expected latency to the destination. In the general case where messages can be destined to any node in the network, this exchangeable message could be a length N vector of expected latencies to all nodes.

The following proposition provides a bound on the expected convergence time of Algorithm 2. The brief proof is provided in Appendix D. The bound depends on the slowest meeting rate between each node and its candidate relay nodes. This is a conservative bound, since in practice, a node only needs to meet the relay nodes to which it actually forwards data under the optimum forwarding rule.

Proposition 1. *The expected convergence time, $E(T_N)$, of Algorithm 2 is bounded as $E(T_N) < \frac{1}{\lambda_{1d}} + \sum_{l=2}^N \frac{l-1}{\min_{\substack{i \in \{1, \dots, l-1\} \\ \lambda_{li} > 0}} \lambda_{li}}.$*

C. Decentralized Approach with No A-priori Knowledge

In part III-B, we assumed that as soon as a node meets another node, it has a perfect knowledge of its meeting rate with that node. In practice, a node will need to estimate its meeting rates with the neighbours and periodically revise the estimation as meetings occur (or fail to occur). Consider an arbitrary pair of nodes that meet each other with rate λ . We denote the i^{th} intermeeting time, which is the time between i^{th} and $i + 1^{\text{th}}$ meetings, by $x_i \geq 0$. For this specific pair of nodes, x_i is an independent sample of an exponentially distributed random variable with parameter λ . Using the maximum likelihood (ML) approach we can estimate the parameter λ after n samples. The

likelihood function $\mathcal{L}(\lambda|x_1, \dots, x_n) = \lambda^n e^{-\lambda \sum_{i=1}^n x_i}$ is maximized by

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^n x_i} \quad (5)$$

Hence, under the exponential model, a node only needs to remember the last time it met its neighbour and the number of times it has met that neighbour. With these two pieces of information, it can update its estimation of the meeting rate ($\hat{\lambda}^n$) from the previously estimated value ($\hat{\lambda}^{n-1}$) using the following equation.

$$\hat{\lambda}^n = \frac{n\hat{\lambda}^{n-1}}{n-1 + \hat{\lambda}^{n-1}x_n} \quad (6)$$

Based on this argument, we develop a more practical version of MinLat in which an arbitrary pair of nodes i and j use their estimated meeting rates $\hat{\lambda}_{ij}$ in their calculations and modify this estimation each time they meet. We refer to this version of MinLat as MinLat-E.

Let t denote the time since the network began operating, and denote by $\tilde{\mathbf{B}}_t$ the decision matrix achieved by MinLat-E at time t . Further, denote by $\tilde{L}_{id,t}(\tilde{\mathbf{B}}_t, i)$ the estimate at node i at time t of the expected latency to the destination, when the forwarding decision matrix is $\tilde{\mathbf{B}}_t$. This estimate differs from that obtained in Algorithm 2, $\hat{L}_{id}(i)$, because the distributed algorithm calculates them using estimated meeting rates $\hat{\lambda}_{i,j}$. The following theorem states that the achieved expected latencies, $L_{id}(\tilde{\mathbf{B}}_t)$ and the estimated expected latencies, $\tilde{L}_{id,t}(\tilde{\mathbf{B}}_t, i)$, converge in probability to the optimum expected latencies $L_{id}(\mathbf{B}^*)$. The proof is provided in Appendix F.

Theorem 4. *For any node in the network, the sequences of estimated and achieved latencies converge to its optimum expected latency in probability, i.e., $\{\tilde{L}_{id,t}(\tilde{\mathbf{B}}_t, i)\} \xrightarrow{p} L_{id}(\mathbf{B}^*)$ and $\{L_{id}(\tilde{\mathbf{B}}_t)\} \xrightarrow{p} L_{id}(\mathbf{B}^*)$. More precisely for any $\epsilon > 0$,*

$$\lim_{t \rightarrow \infty} P(|\tilde{L}_{id,t}(\tilde{\mathbf{B}}_t, i) - L_{id}(\mathbf{B}^*)| < \epsilon) = 1 \quad (7)$$

$$\lim_{t \rightarrow \infty} P(|L_{id}(\tilde{\mathbf{B}}_t) - L_{id}(\mathbf{B}^*)| < \epsilon) = 1 \quad (8)$$

We check the claims of Theorem 4 and investigate the convergence speed of MinLat-E through simulations in Section IV.

IV. SIMULATION RESULTS

In this section, we investigate the efficiency of our proposed approach in modeling and solving the forwarding/routing problem in different opportunistic network scenarios. We first test our algorithms using three different networks to model the contacts between $N = 41$ mobile nodes. The characteristics of the networks are derived from the Infocom05 dataset [40]. This data set is based on an experiment conducted during the IEEE Infocom 2005

conference in Miami where 41 Bluetooth enabled devices (Intel iMote) were carried by attendees for 3 days. The start and end times of each contact between participants were recorded. The average time between node contacts in the Infocom05 dataset is 1.3×10^4 seconds (3.7 hours). In our processing, we only consider the contacts in which both devices recognized each other so that an acknowledged message could be transferred between them.

In the first network, (*Net I*), we construct a contact graph using an evolving undirected network model based on the preferential attachment mechanism. We start with a small fully connected graph of $m_0 = 5$ vertices and add vertices to it one by one until the graph consists of $N = 41$ nodes. At each step, the new vertex is connected to $m = 5$ previously existing vertices. The probability that the new vertex is connected to vertex i is $\frac{k_i}{\sum_j k_j}$ where k_i is the degree of i up to this stage. After building the contact graph, we assign a parameter λ_{ij} to each pair of nodes i and j which are connected in the contact graph and assume that they meet with exponentially distributed intermeeting times with parameter λ_{ij} . We choose the parameters λ_{ij} from a uniform distribution with the same expectation as the average of node meeting rates observed in the Infocom05 dataset.

In *Net II*, we set λ_{ij} to be equal to the inverse of the average intermeeting time between nodes i and j in the Infocom05 dataset. We are interested in the behaviour of the algorithms in relatively sparse networks, so we limit the number of neighbours of each node: node i is only connected to node j in the contact graph if the meeting rate λ_{ij} is among the largest $K = 10$ meeting rates of either node i or node j . In our simulations, the meeting times between nodes i and j for *Net II* are then chosen from an exponential distribution with parameter λ_{ij} . In the third experimental network, *Net III*, we use the actual meeting times recorded in the Infocom05 dataset. The analysis in [37] indicates that the distribution of individual intermeeting times for most pairs of nodes can be approximated reasonably well by an exponential distribution; on the other hand, the aggregate distribution of contact times shows heavy-tailed behaviour and is better approximated using a truncated power distribution [34], [35]. Table I summarizes the properties of the test networks.

TABLE I: Test Network Properties

Net-work	Contact Graph	Parameters	Intermeeting Times	Number of nodes
<i>I</i>	Preferential Attachment	$m_0 = 5$ $m = 5$	Exponential λ : uniform	41
<i>II</i>	Sparsified Infocom	$K = 10$	Exponential λ : data-set	41
<i>III</i>	Sparsified Infocom	$K = 10$	Data-set times	41

As mentioned in Section III, we call our proposed decentralized greedy latency minimization algorithm *MinLat* and refer to its more practical version with meeting rate estimations as *MinLat-E*. In these two algorithms, the decisions that nodes make for future forwarding rules depend on the (estimated) meeting rates, which are derived from the frequency of past contacts between nodes. Thus, MinLat and MinLat-E can be identified as history-based

routing algorithms. In order to evaluate their performance, we compare them with existing history-based routing protocols that can be implemented in a distributed fashion and do not need a-priori knowledge of the network topology. As mentioned in Section I, the fixed point algorithm proposed in [24] is proved to result in the minimum expected latency (which is expected to be the same as the result of our proposed centralized Algorithm 1). However, the proposed algorithm in [24] is centralized and needs to be performed in a control unit where the whole topology of the network is known. The TOUR algorithm proposed in [25] is decentralized, but each node needs perfect a-priori knowledge of its meeting rates with other nodes. Also, the main focus of TOUR is to find the optimum way to make forwarding decisions based on the priorities of messages. We have chosen PRoPHETv2 [32] and MaxProp [15] as the most appropriate candidates for comparison. We also compare to Epidemic routing [10], which is expected to result in a high delivery probability at the expense of high usage of network resources. The parameters of PRoPHETv2 are set to those suggested in [14] and [32], i.e., $P_{init} = 0.75$, $\beta = 0.25$, $\gamma = 0.98$, and time step = 1. In order to put the focus on evaluation of the performance efficiency of forwarding rules and eliminate the effect of the buffer management technique on this performance, we first test the algorithms on ideal network scenarios where the message life times, buffer sizes and data exchanges have no restriction. Therefore for these simulations, the dropping rules proposed for MaxProp in [15] are not applied. We then study the network behaviour when these practical challenges are added to the simulation setups.

We divide the Infocom05 dataset into slots of 12 hours. In each of these time slots, we build networks *I* to *III* using the nodes that are present in that period. The intermeeting time exponential parameters (λ_{ij} s) are estimated based on the meetings that occurred in that specific time slot and networks *I* and *II* are constructed using these estimated parameters. For each of the first four 12-hour periods (the first two days of the conference), we send 1000 messages, spaced by 5-second intervals, from randomly chosen source nodes to a particular destination. We terminate the simulation at the end of the 3-day period, and calculate the fraction of messages successfully delivered by each of the single copy (PRoPHETv2, MinLat) and multi-copy (Epidemic, MaxProp) forwarding algorithms. For each algorithm, we also calculate the average latency of the messages that are delivered by all four algorithms; the average number of hops that messages pass to reach the destination; and the average buffer occupancy of the nodes. For each of the 41 nodes of the network, we calculate the average of performance metrics over the time slots when that node has been chosen as a destination. Figure 1 shows the average and the 95% confidence intervals of the four performance metrics for different destinations in the three test networks using Epidemic, PRoPHETv2, MaxProp, and MinLat forwarding algorithms. There is no restriction on message life time or buffer size that can cause message dropping in these simulations. However, the delivery rates in some cases are less than 1 because the simulations are terminated before all of the generated messages are successfully delivered. We observe that in all the three test networks, MinLat has a better performance than the other existing history-base single copy algorithm,

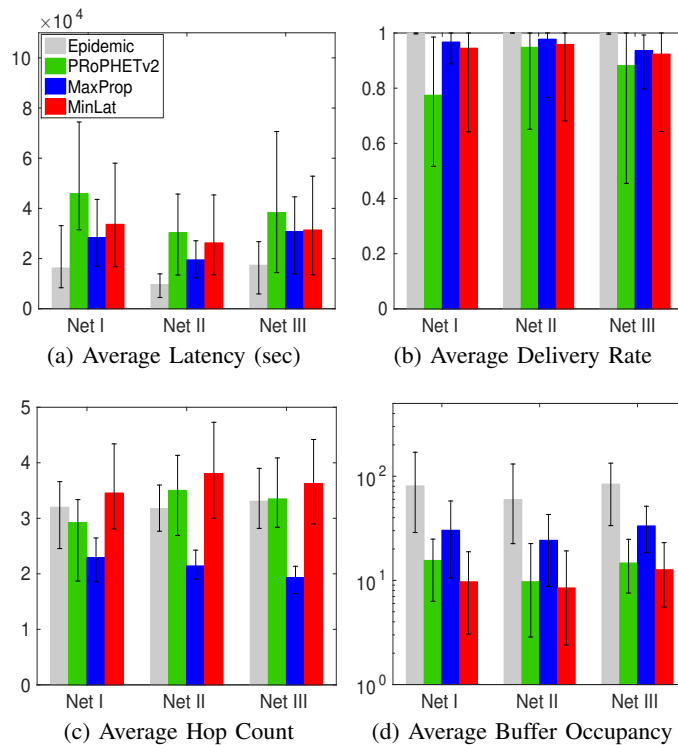


Fig. 1: Comparison Metrics in Test Networks with $N = 41$

PROPHETv2, in terms of delivery rate and average latency. Its performance is also comparable to MaxProp which is a history-based multi-copy algorithm. Also, noting that the scale of the vertical axis of Figure 1d is logarithmic, we see that MinLat occupies much less memory of nodes' buffers in average. For networks *I* and *II*, where the assumption of exponentially distributed intermeeting times holds, delivering a higher rate of messages with lower average latencies than PROPHETv2 is expected from Theorem 3. However, we observe that this result also holds for network *III* where the actual meeting times are used. All algorithms display slightly poorer performance in network *III*; this is probably due to heavy-tailed and non-stationary intermeeting times.

In order to explore how the incorporation of meeting rate estimations in MinLat-E affects the message delivery performance, we conduct further simulations with a different message generation scenario on network *II*. Figure 2 displays the average delivery latency as a function of time for the history-based routing algorithms PROPHET, MaxProp, MinLat, and MinLat-E. The delivery latency values are averaged over 32 simulation runs while the destination node and message generation times are fixed in all the runs. Messages are generated with interarrival time of t seconds at randomly chosen source nodes where t is uniformly distributed in $[0, 5]$. Each point on the curves represents the average latency of the 1500 most recently sent messages. As Figure 2 shows, in all of the three single copy routing algorithms (PROPHET, MinLat, MinLat-E), the average time it takes for a message to be delivered at the destination decreases as time goes by. This decreasing trend is due to the fact that the forwarding rules discovered by nodes improve as the nodes have more contacts and their information concerning their neighbours'

message delivery capabilities becomes more accurate. However, in the multi-copy routing algorithm, MaxProp, the average message delivery time increases in the beginning. In MaxProp, the weights assigned to the links are initialized to be equal, which means that nodes forward messages to more of their neighbours. There is thus a high level of message replication which leads to messages reaching the destination sooner on average. As time passes, the level of replication decreases and the average delivery time increases. We also observe that MinLat-E and MinLat eventually achieve the same average delivery latencies, as expected from Theorem 4. However, the convergence to the optimum point is slower in MinLat-E due to the time it takes for nodes to obtain accurate estimates of their meeting rates.

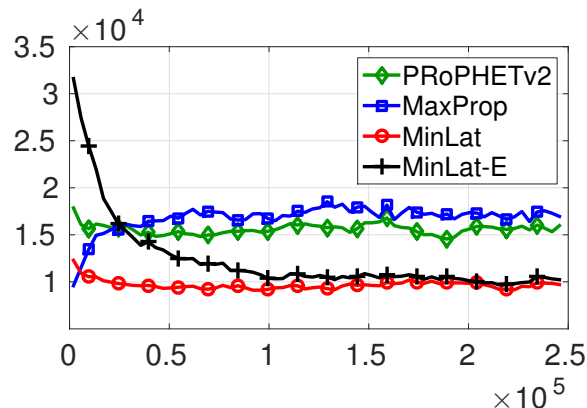


Fig. 2: Evolution of average delivery latency with time (both in seconds).

We examine the performance of the forwarding algorithms in larger networks by extending network I to 100 nodes but with the same average λ parameter for exponential intermeeting times. We also make our model more realistic by adding some practical restrictions to the network model. First, we assume that messages have finite TTL, i.e., a message is discarded when its lifetime exceeds a certain threshold. Figure 3 displays the performance of routing algorithms for different values of TTL varying from 0 to as large as the simulation time (almost 2.5×10^5 seconds). Simulations are run 100 times and in each round, a different destination is randomly chosen from network nodes based on a uniform distribution. The average latency and average hop count are calculated only for the messages that reach the destination.

The simulation results in Figure 3b show that decreasing the TTL has a similar overall effect on all of the algorithms. For larger TTLs, the delivery rate increases, but the buffer occupancy, average latency and hop count also increase. MinLat outperforms PRoPHET and MaxProp in terms of delivery rate, average latency, and buffer occupancy even in a scenario with a restricted message life time. Although intermeeting times are exponentially distributed and the contact graph is based on preferential attachment, in this larger network of 100 nodes, PRoPHET cannot reach 100 percent delivery rate even without any restriction on TTL.

The next step towards a more realistic network model is to consider limits on buffer size. In the next set of

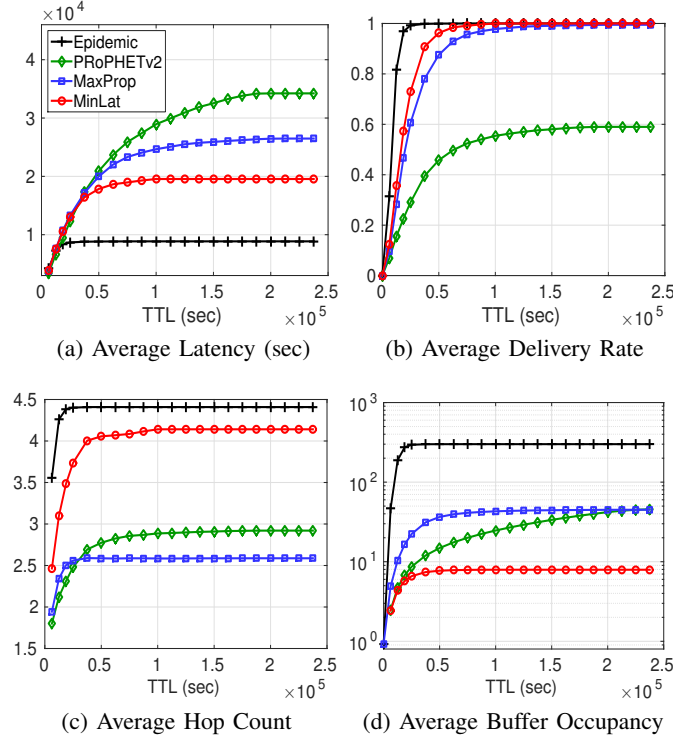


Fig. 3: Effect of TTL on Performance Metrics

simulations, we assume that TTL is 10^5 seconds so that all algorithms reach their best possible delivery rate. We also assume that each node has a limited capacity for keeping the messages. When the buffer occupancy of a node reaches its limit, messages from other nodes are not forwarded to it. Moreover, any generated messages at the fully occupied node are immediately dropped. Figure 4 shows the performance of algorithms for buffer sizes in the range of 0 to 1250 messages. We see that increasing the buffer size improves the delivery rate for all algorithms. It also reduces the average latency because as the buffer size increases, nodes can more frequently follow their optimum forwarding rules.

Finally, we assume that the contact duration is limited so that the number of messages during any meeting is restricted by an *exchange limit*. We set the buffer size to 1000 messages so that all the algorithms reach their best possible delivery rate. This buffer size implies 50 MB of node memory if each message is 50 KB. We check the effect of varying the exchange limit on the network performance. Figure 5 shows the four comparison metrics for exchange limits in the interval 0 to 500 messages. As we see in the figure, MinLat cannot achieve the optimum average latency for some values of exchange limit, but it still has the best performance in terms of buffer occupancy.

The simulation results displayed in Figures 3-5 illustrate that although MinLat has been designed for an ideal network model, it has an acceptable performance when we impose realistic conditions such as finite TTL, buffer size, and exchange limit. Our final set of simulations explores the impact of including the recursive maximum likelihood

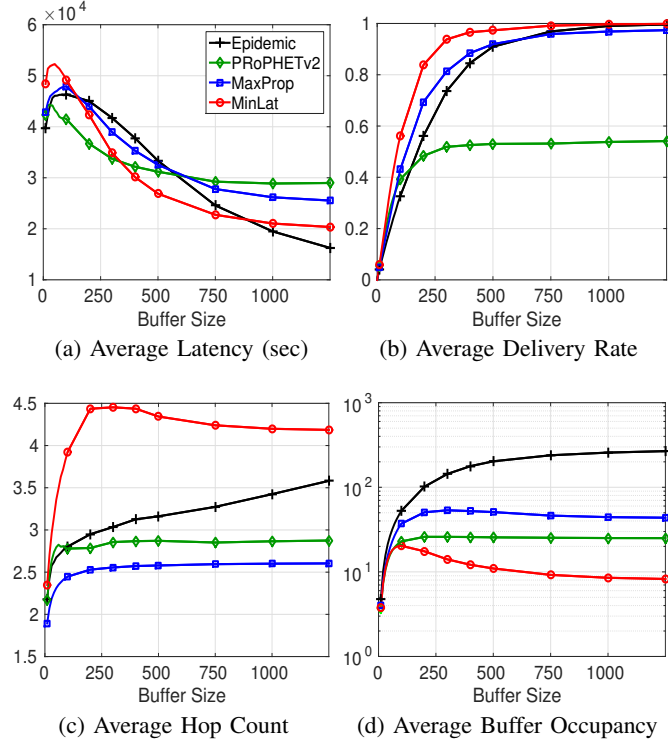


Fig. 4: Effect of Buffer Size on Performance Metrics

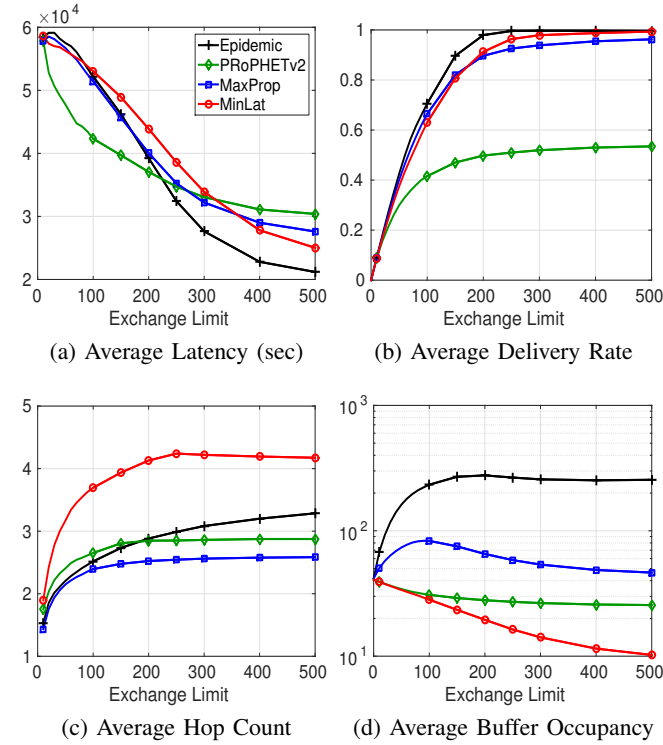


Fig. 5: Effect of Exchange Limit on Performance Metrics

estimation of meeting rates. We conduct experiments using both the centralized and decentralized algorithms operating on the extension of network I to 100 nodes. We select the λ parameters of intermeeting times from

a uniform distribution $U[0, 0.01]$. The destination node is randomly chosen from the N nodes of the network based on a uniform distribution and is fixed throughout the simulation. We run the simulation for 5×10^4 seconds. We examine the error terms in Theorem 4, averaging over all nodes. Figure 6a shows the average absolute difference between the estimated expected latencies and the true minimum latencies, i.e., $\frac{1}{N} \sum_k |\tilde{L}_{kd,t}(\tilde{\mathbf{B}}_t, k) - L_{kd}(\mathbf{B}^*)|$. Figure 6b shows the difference between the achieved average latencies and the true minimum latencies, i.e., $\frac{1}{N} \sum_k |L_{kd}(\tilde{\mathbf{B}}_t) - L_{kd}(\mathbf{B}^*)|$. Figure 6 indicates that the decentralized algorithm achieves almost the same estimation error as the centralized algorithm, suggesting that the limiting effect is the convergence of the meeting rate estimates rather than the dissemination of latency estimates through the network. As expected from Theorem 4, we see that the estimated and achieved errors both decay to zero as the time goes by (for achieved latencies, it is almost zero for all nodes of the network after $t = 7000$ seconds).

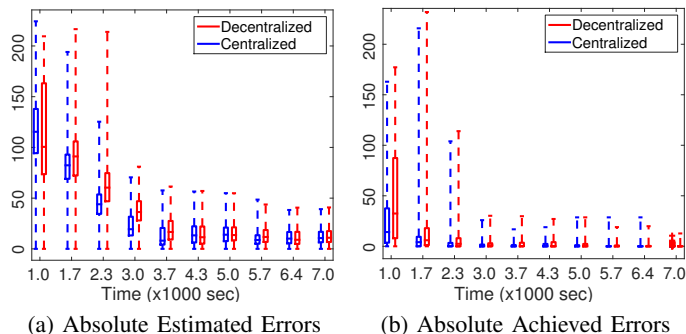


Fig. 6: Absolute Errors of Estimated and Achieved Latencies. The central mark of each box is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the highest values not considered as outliers, and outliers are plotted individually.

Comparing Figures 6a and 6b shows that for both centralized and decentralized scenarios, the average difference between the achieved latency and the minimum latency is less than the average difference between the estimated latency and the minimum latency. This is expected, because the estimated latencies are based on incorrect decision matrices and estimated meeting rates, whereas the achieved latencies are derived from the actual meeting rates. The results suggest that even when there remains substantial inaccuracy in the expected latency estimates (e.g. at time $t = 3000$ seconds), the algorithm identifies a close-to-optimal forwarding decision matrix.

V. CONCLUSION

In this paper, we used an analytical framework to model the opportunistic data transfer among mobile devices in Delay Tolerant Networks. In our model, the random intermeeting times of nodes are assumed to be independent and exponentially distributed. We formulated the routing/forwarding problem as an optimization problem in which the goal is to minimize the sum of expected latencies from all nodes of the network to a particular destination. We proved that the solution of this problem is binary, i.e., when an arbitrary node meets any other node in the

network, its optimum forwarding rule dictates either to always forward its messages to the encountered node or to never forward any messages to it. We also showed that the solution of this optimization problem minimizes the expected latency from each node of the network to the destination as well. Based on these results, we proposed centralized and distributed versions of an algorithm to find the optimum forwarding decision rules and proved that each of these algorithms result in the same solution. In order to evaluate the performance efficiency of the suggested algorithms in different synthetic and real-world networks, we chose four performance metrics as comparison metrics and compared our proposed decentralized algorithm (MinLat) with the most similar existing approaches. In order to evaluate the performance of MinLat in more realistic scenarios, we conducted simulations in larger networks with practical constraints like limited message life (TTL), buffer size and message exchange.

One of the main contributions of this work is relaxing the condition of having complete knowledge of meeting rates at each node for making the forwarding decisions. We used a recursive maximum likelihood procedure (MinLat-E) to learn the meeting rates online and proved its convergence in probability to the same optimal solution. The validity of this theoretical result was assessed through simulations. Moreover, we compared the convergence speed of the proposed centralized and decentralized algorithms when the meeting rates are estimated online. The simulation results show that the decentralized algorithm has almost the same convergence rate as the centralized algorithm, even though the network topology is not known at individual nodes.

In future work, we aim to explore the effect of time-varying meeting rates between nodes, which would motivate the use of filters to track the rates. We also hope to examine whether it is possible to derive similar expressions for expected latencies and optimal forwarding rules for cases when the intermeeting times are not exponentially distributed or are correlated.

APPENDIX A

PROOF OF LEMMA 1

When node i commences in its routing of a packet, it must first wait a time T_w before it meets one of its neighbors. The amount of time before node i meets a specific neighbor j is an exponentially distributed random variable with parameter λ_{ij} . The time T_w is equal to the minimum of the exponentially distributed random variables corresponding to all neighbours $k \in \mathcal{S}_i$ and its expected value is

$$E(T_w) = \frac{1}{\sum_{k \in \mathcal{S}_i} \lambda_{ik}} \quad (9)$$

The probability that j is the first node that i meets is $\frac{\lambda_{ij}}{\sum_{k \in \mathcal{S}_i} \lambda_{ik}}$. Hence $L_{id}(\mathbf{P})$ is

$$L_{id}(\mathbf{P}) = E(T_w) + \sum_{j \in \mathcal{S}_i} \frac{\lambda_{ij}}{\sum_{k \in \mathcal{S}_i} \lambda_{ik}} [p_{ij} L_{jd}(\mathbf{P}) + (1 - p_{ij}) L_{id}(\mathbf{P})] \quad (10)$$

The last term in (10) follows from the memoryless property of the distributions. Substituting (9) into (10) leads to (1). \square

APPENDIX B

PROOF OF THEOREM 1

Let us assume that the nodes, excluding d , are labelled in ascending order of their expected latency under \mathbf{P}^* , i.e., $L_{1d}(\mathbf{P}^*) < L_{2d}(\mathbf{P}^*) < \dots < L_{N-1d}(\mathbf{P}^*)$. For a given matrix \mathbf{P} , we denote by $\mathbf{P}_{\bar{i}}$ all rows of \mathbf{P} except i . If we fix \mathbf{p}_i and L_{kd} for $k \in \mathcal{S}_i, k \neq j$ for some $j \in \mathcal{S}_i$, then L_{id} is monotonically increasing with respect to L_{jd} (see (1)). This implies that if we commence with any \mathbf{P} and change only \mathbf{p}_j to decrease L_{jd} , then all other L_{id} such that $j \in \mathcal{S}_i$ either decrease or remain the same. The matrix \mathbf{P}^* must therefore satisfy $\mathbf{p}_j^* = \arg \min L_{jd}(\mathbf{P}_j^*, \mathbf{p}_j)$ for all j . Otherwise we could choose an alternative \mathbf{p}'_j that reduces L_{jd} and hence achieves $\mathcal{U}(\mathbf{P}') < \mathcal{U}(\mathbf{P}^*)$.

We can examine the partial derivative of L_{id} with respect to p_{ij} at $\mathbf{P}' = (\mathbf{P}_{\bar{i}}^*, \mathbf{p}_i)$:

$$\frac{\partial L_{id}}{\partial p_{ij}} = \frac{\lambda_{ij}[\sum_{k \in \mathcal{S}_i} \lambda_{ik} p_{ik} (L_{jd}(\mathbf{P}') - L_{kd}(\mathbf{P}')) - 1]}{(\sum_{k \in \mathcal{S}_i} \lambda_{ik} p_{ik})^2} \quad (11)$$

This derivative has the same sign as: $L_{jd}(\mathbf{P}') - \frac{1 + \sum_{k \in \mathcal{S}_i} \lambda_{ik} p_{ik} L_{kd}(\mathbf{P}')}{\sum_{k \in \mathcal{S}_i} \lambda_{ik} p_{ik}}$, or equivalently $L_{jd}(\mathbf{P}') - L_{id}(\mathbf{P}')$. This expression for the derivative, together with the requirement that $\mathbf{p}_i^* = \arg \min L_{id}(\mathbf{P}_{\bar{i}}^*, \mathbf{p}_i)$, implies that $\mathbf{p}_{ij}^* = 0$ if $L_{id}(\mathbf{P}^*) < L_{jd}(\mathbf{P}^*)$ and $\mathbf{p}_{ij}^* = 1$ if $L_{id}(\mathbf{P}^*) > L_{jd}(\mathbf{P}^*)$. Our assumption that the solution is unique implies that $L_{id}(\mathbf{P}^*) \neq L_{jd}(\mathbf{P}^*)$. Otherwise, from (10), it is clear that we could choose any p_{ij}^* between 0 and 1 and achieve the same $L_{id}(\mathbf{P}^*)$, without affecting any other $L_{jd}(\mathbf{P}^*)$. This establishes statement (1) of the theorem.

Although we have established that $L_{id}(\mathbf{P}_{\bar{i}}^*, \mathbf{p}_i^*) = \min L_{id}(\mathbf{P}_{\bar{i}}^*, \mathbf{p}_i)$, we have not yet shown that \mathbf{P}^* globally minimizes L_{id} . We establish this by contradiction. Suppose \mathbf{P}^* does not minimize the expected latency for some non-empty set of nodes $\mathcal{N}' \subset \mathcal{N}$. In other words, denoting the minimum expected latency achieved via the minimization in (2) for node i by L_{id}^* , we have

$$\forall i \in \mathcal{N}' : \quad L_{id}^* < L_{id}(\mathbf{P}^*) \quad (12)$$

Let node s be the node in \mathcal{N}' such that $L_{sd}^* < L_{kd}^*$ for all $k \in \mathcal{N}', k \neq s$. Denote by ℓ the ranking of the node with greatest expected latency under \mathbf{P}^* such that $L_{\ell d}(\mathbf{P}^*) < L_{sd}^*$. Based on the discussion above, for each node $i \in \{1, 2, \dots, \ell\}$, $p_{ik}^* = 0$ for all $k > \ell$ and hence $p_{is}^* = 0$. Node s must have at least one neighbour in the set $\{d, 1, 2, \dots, \ell\}$. Otherwise, it could not achieve an expected latency under \mathbf{P}^* that is less than all nodes $\ell + 1, \dots, N - 1$ (observe from (1) that $L_{sd}(\mathbf{P}^*) > \min_{p_{sj} > 0} L_{jd}(\mathbf{P}^*)$).

The matrix \mathbf{P}' that achieves the minimum L_{sd}^* must satisfy $p'_{sk} = 0$ for all $k \in \mathcal{N}'$, since for any matrix \mathbf{P}' we have $L_{kd}(\mathbf{P}') \geq L_{kd}^* > L_{sd}^*$. We also have $p'_{sk} = 1$ for $k \in \mathcal{S}_s \cap \{d, 1, 2, \dots, \ell\}$ if $L_{kd}(\mathbf{P}') < L_{sd}(\mathbf{P}')$. For a fixed

choice of \mathbf{p}'_s the value $L_{sd}(\mathbf{P}')$ decreases if we can reduce $L_{kd}(\mathbf{P}')$ for any k such that $p'_{sk} = 1$. The matrix \mathbf{P}^* minimizes L_{kd} for all $k \in \{1, 2, \dots, \ell\}$, implying that $\mathbf{p}'_k = \mathbf{p}^*_k$ for all $k \in \{1, 2, \dots, \ell\}$. Since $L_{kd}(\mathbf{P}') = L_{kd}(\mathbf{P}^*)$ for all $k \in \mathcal{S}_s \cap \{d, 1, 2, \dots, \ell\}$, it follows that $\mathbf{p}'_s = \mathbf{p}^*_s$. For node s , the values of \mathbf{p}'_j for $j \notin \{1, 2, \dots, \ell, s\}$ have no impact on L_{sd} , so we have $L_{sd}(\mathbf{P}^*) = L_{sd}(\mathbf{P}') = L_{sd}^*$. This contradicts the original assumption that \mathbf{P}^* does not minimize the latency for all nodes $s \in \mathcal{N}'$, and thus establishes statement (2) of the theorem. \square

APPENDIX C

PROOF OF THEOREM 2

We observe that for all $i \in \mathcal{N}$, $G_{id} \geq L_{id}^*$ (since the optimizations are the same). Based on Theorem 1 and its proof, the equality holds only if $j \in \mathcal{A}$ for all $j \in \mathcal{S}_i$ such that $L_{jd}^* < L_{id}^*$. The statements in the theorem follow based on an induction argument.

Suppose, without loss of generality, that the nodes are labelled in ascending order of expected latency under \mathbf{B}^* . For node 1, the only neighbour with lower expected latency is the destination. In iteration 1, the destination is included in \mathcal{A} and must be in \mathcal{S}_1 . Recall that $L_{1d}^* > \min_{b_{1j}=1} L_{jd}^*$. Node 1 has the minimum expected latency according to the chosen labelling and Theorem 1, except for the destination itself. The relationship thus implies that $d \in \mathcal{S}_1$. We therefore have $G_{1d} = L_{1d}^* < L_{jd}^* \leq G_{jd}$, and node 1 is selected to be added to \mathcal{A} , with $b_{1d} = 1$ and $b_{jd} = 0$ for all $j \neq d$. Statements 1a)-c) in the theorem clearly hold after one iteration, i.e., after the addition of node 1 to \mathcal{A} .

Assume the same statements hold after the addition of node $k-1$ to \mathcal{A} . Then, for node k we must have $j \in \mathcal{A}$ for all $j \in \mathcal{S}_k$ such that $L_{jd}^* < L_{kd}^*$. Again this implies that $G_{kd} = L_{kd}^* < L_{jd}^* \leq G_{jd}$ for all $j > k$. Thus, node k is correctly selected for addition to \mathcal{A} and the statements 1a)-c) hold at the end of iteration k .

It follows that the statements hold for all iterations of the algorithm, and after completion, when $\mathcal{A} = \mathcal{N}$, the second statement follows. \square

APPENDIX D

PROOF OF THEOREM 3 AND PROPOSITION 1

A. Proof of Theorem 3

Assume that the nodes are labelled in order of ascending expected latency under \mathbf{B}^* , i.e., $L_{1d}(\mathbf{B}^*) \leq L_{2d}(\mathbf{B}^*) \leq \dots \leq L_{N-1d}(\mathbf{B}^*)$. Denote by T_1 the moment of time at which node 1 meets the destination node. For $k = 2, \dots, N$ denote by T_k the earliest time by which node k has met all nodes in the set $\{1, \dots, k-1\} \cap \mathcal{S}_k$ in the time period $(T_{k-1}, T_k]$. Due to the assumption that the inter-meeting times are exponentially distributed, T_N is finite with probability 1.

At T_1 , node 1 learns its meeting rate with the destination (λ_{1d}). Since the estimated latencies are initialized to ∞ and due to the update equations in Algorithm 2, the estimation that node 1 has at T_1 of the latencies of its neighbors $i \in \mathcal{S}_1$ are upper-bounds, i.e. $\hat{L}_{id}(1) \geq L_{id}(\mathbf{B}^*)$. As discussed in the proof of the previous theorems, the minimizer \mathbf{b}_1^* has $b_{1d} = 1$ and $b_{1j} = 0$ for all $j \neq d$. At time T_1 , since the term involving d in the update equation of Algorithm 2 has its minimum value, the vector $\mathbf{m}_1^* = \mathbf{b}_1^*$ identifies the same minimum latency $\hat{L}_{1d}(1) = L_{1d}(\mathbf{B}^*)$. Hence, immediately after time T_1 we are guaranteed that $\mathbf{b}_1 = \mathbf{b}_1^*$.

At T_k , node k is aware of the minimum expected latencies $\hat{L}_{sd}(k) = L_{sd}(\mathbf{B}^*)$ for the nodes in the set $\mathcal{V}_k = \{d, 1, \dots, k\} \cap \mathcal{S}_k$. All other expected latencies are upper bounds, i.e. $\hat{L}_{jd}(k) \geq L_{jd}(\mathbf{B}^*)$ for $j \notin \mathcal{V}_k$. The solution \mathbf{b}_k^* takes value 1 only for nodes in \mathcal{V}_k . The minimizer \mathbf{m}_k^* at time T_k is thus equal to \mathbf{b}_k^* and achieves $\hat{L}_{kd}(k) = L_{kd}(\mathbf{B}^*)$. Therefore, immediately after T_k we will have $\mathbf{b}_k = \mathbf{b}_k^*$. This argument applies until just after T_N , at which point we have $\mathbf{B} = \mathbf{B}^*$. Since T_N is finite with probability 1, the statement of the theorem follows.

B. Proof of Proposition 1

Algorithm 2 has converged when all of the nodes have met all of their relay candidates and have identified their optimum forwarding rules. Thus, $E(T_N)$, where T_N defined above in the proof of Theorem 3, is the expected convergence time. Considering the worst case where an arbitrary node $k > 1$ is connected to all the nodes in the set $\{1, \dots, k-1\}$, we have

$$E(T_k - T_{k-1}) = E\left(\max_{i \in \{1, \dots, k-1\}} x_i\right) \quad (13)$$

where x_i denotes the intermeeting time of node k with its neighbour i and follows an exponential distribution with parameter λ_{ki} . Using a standard result for the expected value of the maximum of non-identical independent exponential random variables, we have

$$E(T_k - T_{k-1}) = \sum_{i=1}^{k-1} \frac{1}{\lambda_{ki}} - \sum_{i=1}^{k-1} \sum_{j=i+1}^{k-1} \frac{1}{\lambda_{ki} + \lambda_{kj}} + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k-1} \sum_{l=j+1}^{k-1} \frac{1}{\lambda_{ki} + \lambda_{kj} + \lambda_{kl}} - \dots \quad (14)$$

An upper bound on this value is $E(T_k - T_{k-1}) < \sum_{i=1}^{k-1} \frac{1}{\lambda_{ki}} < \frac{k-1}{\min_i \lambda_{ki}}$. Therefore, an upperbound on the convergence time of Algorithm 2 is $E(T_N) < \frac{1}{\lambda_{1d}} + \sum_{l=2}^N \frac{l-1}{\min_{i \in \{1, \dots, l-1\}} \lambda_{li}}$. \square

APPENDIX E

PROOF OF LEMMA 2

Let's return to considering the probability decision variable vector \mathbf{p} instead of binary decision variable vector. (We still expect the optimal solution to be of the binary form). Without loss of generality, we relabel the nodes $k \in \mathcal{S}_i$ by labels $1, \dots, |\mathcal{S}_i|$. The optimization problem that node i tries to solve in Algorithm 2 follows this general form

which is known as linear fractional optimization problems:

$$\begin{aligned} \widehat{L}_{id}(i) &= \min_{\mathbf{p}^i} \frac{\mathbf{c}^T \mathbf{p}^i + \alpha}{\mathbf{d}^T \mathbf{p}^i + \beta} \\ \text{subject to} \quad & \mathbf{A} \mathbf{p}^i \leq \mathbf{b} \end{aligned} \quad (15)$$

where $\mathbf{p}_{|\mathcal{S}_i| \times 1}^i = [p_{i1}, \dots, p_{i|\mathcal{S}_i|}]^T$, $\alpha = 1$, $\mathbf{c}_{|\mathcal{S}_i| \times 1} = [\lambda_{i1} \widehat{L}_{1d}(i), \dots, \lambda_{i|\mathcal{S}_i|} \widehat{L}_{|\mathcal{S}_i|d}(i)]^T$, $\beta = 0$, $\mathbf{d}_{|\mathcal{S}_i| \times 1} = [\lambda_{i1}, \dots, \lambda_{i|\mathcal{S}_i|}]^T$, $\mathbf{b}_{2|\mathcal{S}_i| \times 1} = [1, \dots, 1, 0, \dots, 0]^T$, and the elements of the matrix $\mathbf{A}_{2|\mathcal{S}_i| \times |\mathcal{S}_i|}$ are

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } i < |\mathcal{S}_i| \quad \& \quad j = i \\ -1 & \text{if } i > |\mathcal{S}_i| \quad \& \quad j = i - |\mathcal{S}_i| \\ 0 & \text{otherwise} \end{cases}$$

After applying the following parameter changes (the Charnes-Cooper transformation), we have

$$\mathbf{x} = \frac{1}{\mathbf{d}^T \mathbf{p}^i} \mathbf{p}^i \quad y = \frac{1}{\mathbf{d}^T \mathbf{p}^i} \quad (16)$$

and the optimization problem 15 converts to

$$\begin{aligned} \min_{\mathbf{x}, y} \quad & \mathbf{c}^T \mathbf{x} + \alpha y \\ \text{subject to} \quad & \mathbf{d}^T \mathbf{x} = 1 \\ & \mathbf{A} \mathbf{x} \leq \mathbf{b} y \\ & y \geq 0 \end{aligned} \quad (17)$$

which is a linear optimization problem and can be solved using Linear Programming (LP) solution methods. \square

APPENDIX F

PROOF OF THEOREM 4

We first prove the statement of the theorem for estimated latencies. Without loss of generality, we relabel the nodes such that $L_{1d}(\mathbf{B}^*) < L_{2d}(\mathbf{B}^*) < \dots < L_{Nd}(\mathbf{B}^*)$. At node i , MinLat-E forms estimates of the meeting rates with the contact graph neighbours, $\widehat{\lambda}_{ik}$ for $k \in \mathcal{S}_i$, and the expected latencies from each neighbour $\widetilde{L}_{kd}(\widetilde{\mathbf{B}}_t, i)$. The result is a sequence of random variables $\{\widetilde{L}_{id}(\widetilde{\mathbf{B}}_t, i)\}$, with a variable being added to the sequence each time node i meets another node. [41] shows that the maximum likelihood estimator of the meeting rates is consistent, i.e., $\forall i, j \in \mathcal{N}, \widehat{\lambda}_{ij} \xrightarrow{p} \lambda_{ij}$. More precisely,

$$\forall \epsilon_{ij} > 0 : \lim_{t \rightarrow \infty} P(|\widehat{\lambda}_{ij} - \lambda_{ij}| < \epsilon_{ij}) = 1 \quad (18)$$

Equivalently, writing $\epsilon_{ij} = \epsilon_0 \lambda_{ij}$, we have for any $\delta > 0$ and $\epsilon_0 > 0$, there exists a $t_0 > 0$ such that for all $t > t_0$:

$$P((1 - \epsilon_0)\lambda_{ij} < \hat{\lambda}_{ij,t} < (1 + \epsilon_0)\lambda_{ij}) > (1 - \delta) \quad (19)$$

where $\hat{\lambda}_{ij,t}$ denotes the estimate of the meeting rate between nodes i and j at time t .

We show that for any set of meeting rates $\{\lambda_{ij}\}_{i,j \in \mathcal{N}}$, there exists an $\epsilon = \epsilon_0$ for which the optimum forwarding decision matrix in MinLat-E ($\tilde{\mathbf{B}}_t$) is the same as the optimum forwarding decision matrix in MinLat (\mathbf{B}^*) with desirably high probability. In order to do so, we find upper and lower bounds (that apply with high probability) on the estimated expected latencies for the optimal decision matrices identified by both MinLat and MinLat-E. We first demonstrate a relationship between the estimated and true expected latencies that would hold if the nodes employed the optimal decision matrix \mathbf{B}^* . We show that there exists a $t_{\delta,i}$ such that for each node $i \in \mathcal{N}$, with probability greater than $1 - \delta$, for any positive δ , we have for all $t > t_{\delta,i}$:

$$\frac{(1 - \epsilon_0)^{i-1}}{(1 + \epsilon_0)^i} L_{id}(\mathbf{B}^*) < \tilde{L}_{id,t}(\mathbf{B}^*, i) < \frac{(1 + \epsilon_0)^{i-1}}{(1 - \epsilon_0)^i} L_{id}(\mathbf{B}^*) \quad (20)$$

We derive (20) by induction. From the arguments made in Theorem 3, we know that under \mathbf{B}^* an arbitrary node i will not forward any messages to a node that does not belong to the set $\{1, \dots, i-1, d\}$. For $i = 1$, after time $t_{\delta,1}$ such that (19) holds, we have, with probability greater than $1 - \delta$ for $t > t_{\delta,1}$:

$$\frac{L_{1d}(\mathbf{B}^*)}{1 + \epsilon_0} = \frac{1}{1 + \epsilon_0} \frac{1}{\lambda_{1d}} < \tilde{L}_{1d,t}(\mathbf{B}^*, 1) = \frac{1}{\hat{\lambda}_{1d,t}} < \frac{1}{1 - \epsilon_0} \frac{1}{\lambda_{1d}} = \frac{L_{1d}(\mathbf{B}^*)}{1 - \epsilon_0} \quad (21)$$

Suppose (20) holds for all the nodes $1, \dots, k-1$. Denote by $t_{jk} > t$ the last meeting between node j and k that occurs subsequent to $t_{\delta,k-1}$ but prior to a considered time t . Then we can identify a $t_{\delta,k}^u$ so that the following relationship holds with probability greater than $1 - \delta$ for $t > t_{\delta,k}^u$:

$$\tilde{L}_{kd,t}(\mathbf{B}^*, k) = \frac{1}{\sum_{j \in \mathcal{S}_k} b_{kj}^* \hat{\lambda}_{kj}} (1 + \sum_{j \in \mathcal{S}_k} b_{kj}^* \hat{\lambda}_{kj} \tilde{L}_{jd,t_{jk}}(\mathbf{B}^*, j)) \quad (22a)$$

$$< \left(\frac{1}{1 - \epsilon_0} \frac{1}{\sum_{j \in \mathcal{S}_k} b_{kj}^* \lambda_{kj}} \right) (1 + \sum_{j \in \mathcal{S}_k} b_{kj}^* (1 + \epsilon_0) \frac{(1 + \epsilon_0)^{k-2}}{(1 - \epsilon_0)^{k-1}} L_{jd}(\mathbf{B}^*)) \quad (22b)$$

$$< \frac{(1 + \epsilon_0)^{k-1}}{(1 - \epsilon_0)^k} L_{kd}(\mathbf{B}^*) \quad (22c)$$

Here we have chosen $t_{\delta,k}^u$ to be sufficiently large such that the inequality on the second line holds with probability exceeding $1 - \delta$. Similarly we can identify a $t_{\delta,k}^\ell$ so that the lower bound in (20) holds with probability greater than $1 - \delta$ for $t > t_{\delta,k}^\ell$. By taking $t_{\delta,k} = \max\{t_{\delta,k}^\ell, t_{\delta,k}^u\}$, we see that (20) holds for node k as well, and by induction, holds for all nodes $i \in \mathcal{N}$.

We now turn our attention to the forwarding matrix determined by MinLat-E ($\tilde{\mathbf{B}}_t$). We first consider a scenario

where the estimates of the meeting rates are frozen after a certain time t_f . The minimization procedure in MinLat-E is the same as in MinLat, but operates on the estimates of the meeting rates. If these estimates are held constant, then the results in Theorems 1-3 apply, with the substitution of $\hat{\lambda}_{ij}$ anywhere we make use of λ_{ij} . With probability 1, the optimization algorithm will thus converge after a finite time t' , and the estimated expected latencies will be consistent across the network. Hence, there exists a labeling $\{\hat{1}, \hat{2}, \dots, \hat{N}\}$ for which $\tilde{L}_{\hat{1}d}(\tilde{\mathbf{B}}_{t_f+t'}, \hat{1}) < \tilde{L}_{\hat{2}d}(\tilde{\mathbf{B}}_{t_f+t'}, \hat{2}) < \dots < \tilde{L}_{\hat{N}d}(\tilde{\mathbf{B}}_{t_f+t'}, \hat{N})$. We can now employ the same argument that was used above for \mathbf{B}^* to determine that there is a finite time $t_{\delta, \hat{k}}$ such that the following bound holds for all $\hat{k} \in \mathcal{N}$ with probability greater than $1 - \delta$ for all $t > t_{\delta, \hat{k}}$.

$$\frac{(1 - \epsilon_0)^{\hat{k}-1}}{(1 + \epsilon_0)^{\hat{k}}} L_{\hat{k}d}(\tilde{\mathbf{B}}_t) < \tilde{L}_{\hat{k}d}(\tilde{\mathbf{B}}_t, \hat{k}) < \frac{(1 + \epsilon_0)^{\hat{k}-1}}{(1 - \epsilon_0)^{\hat{k}}} L_{\hat{k}d}(\tilde{\mathbf{B}}_t) \quad (23)$$

In the actual MinLat-E algorithm, the meeting rates $\hat{\lambda}$ are not frozen after t_0 , but continue to be updated as more meetings occur. This only results in the probabilistic bounds on $\hat{\lambda}$ being tighter, and hence can only tighten the bounds on $L_{\hat{k}d}(\tilde{\mathbf{B}}_t, \hat{k})$.

To avoid having node-specific bounds on the accuracy of the estimates, we can rewrite the bounds as:

$$\frac{(1 - \epsilon_0)^{N-1}}{(1 + \epsilon_0)^N} L_{kd}(\mathbf{B}) < \tilde{L}_{kd}(\mathbf{B}, k) < \frac{(1 + \epsilon_0)^{N-1}}{(1 - \epsilon_0)^N} L_{kd}(\mathbf{B}) \quad (24)$$

This bound holds for both $\mathbf{B} = \mathbf{B}^*$ and $\mathbf{B} = \tilde{\mathbf{B}}_t$ with probability at least $1 - \delta_0$ after some time t_0 .

Our goal is to show that there exists a moment of time after which $\tilde{\mathbf{B}}_t = \mathbf{B}^*$ is true with desirably high probability. We can accomplish this by showing that there exists an ϵ_0 (and thus an associated time t_0) for which the upper-bound on $\tilde{L}_{id}(\mathbf{B}^*, i)$ is less than the lower-bound on $\tilde{L}_{id}(\tilde{\mathbf{B}}_t, i)$ for any $\tilde{\mathbf{B}}_t \neq \mathbf{B}^*$ for all $t > t_0$. If this is the case, then with probability exceeding $1 - \delta_0$, the optimization procedure that derives $\tilde{\mathbf{B}}_t$ will set it to \mathbf{B}^* , because it minimizes the estimated latencies. Hence, ϵ_0 should satisfy

$$L_{id}(\mathbf{B}^*) < \left(\frac{1 - \epsilon_0}{1 + \epsilon_0}\right)^{2N-1} L_{id}(\tilde{\mathbf{B}}_t), \quad (25)$$

which leads to

$$\epsilon_0 < \frac{e^{\frac{\ln(K)}{2N-1}} - 1}{e^{\frac{\ln(K)}{2N-1}} + 1}, \quad (26)$$

where $K = \frac{\min_{\mathbf{B} \neq \mathbf{B}^*} L_{id}(\mathbf{B})}{L_{id}(\mathbf{B}^*)}$.

Now that we have established that after a finite amount of time $\tilde{\mathbf{B}}_t = \mathbf{B}^*$ occurs with a probability desirably close to one, we can show that $\{\tilde{L}_{id}(\tilde{\mathbf{B}}_t, i)\} \xrightarrow{p} L_{id}(\mathbf{B}^*)$ for any $i \in \mathcal{N}$ using the following properties:

- 1) If $X_n \xrightarrow{p(\text{or } d)} X$, then $g(X_n) \xrightarrow{p(\text{or } d)} g(X)$ (Continuous Mapping Theorem);
- 2) If $X_n \xrightarrow{p} X$, then $X_n \xrightarrow{d} X$;
- 3) If $X_n \xrightarrow{d} c \in \mathbb{R}$, then $X_n \xrightarrow{p} X$;

4) If $X_n \xrightarrow{d} X$ and $Y_n \xrightarrow{d} c \in \mathbb{R}$, then $g(X_n, Y_n) \xrightarrow{d} g(X, c)$ (Slutsky's Theorem),

where \xrightarrow{d} denotes the convergence in distribution and $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is an arbitrary continuous function.

Again consider the node labelling such that $L_{1d}(\mathbf{B}^*) < L_{2d}(\mathbf{B}^*) < \dots < L_{Nd}(\mathbf{B}^*)$. For node 1, $\{\tilde{L}_{1d}(\tilde{\mathbf{B}}_t, 1) = \frac{1}{\lambda_{1d}}\} \xrightarrow{p} L_{1d}(\mathbf{B}^*) = \frac{1}{\lambda_{1d}}$ which is obvious from property 1. For any node $k > 1$, if $\{\tilde{L}_{jd}(\tilde{\mathbf{B}}_t, j)\} \xrightarrow{p} L_{jd}(\mathbf{B}^*)$ holds for any $j \in \{1, \dots, k-1\}$, then due to properties 2-4 we have,

$$\begin{aligned} \forall j \in \{1, \dots, k-1, d\} \cap \mathcal{S}_k : \\ \{\hat{\lambda}_{kj} \tilde{L}_{jd}(\tilde{\mathbf{B}}_t, k)\} \xrightarrow{d} \lambda_{kj} L_{jd}(\mathbf{B}^*) \in \mathbb{R} , \end{aligned} \quad (27)$$

and

$$\{\sum_{j \in \{1, \dots, k-1, d\}} \hat{b}_{kj}^* \hat{\lambda}_{kj}\} \xrightarrow{p} \sum_{j \in \{1, \dots, k-1, d\}} b_{kj}^* \lambda_{kj} . \quad (28)$$

Property 1 in combination with (27) and (28) results in the statement of the theorem.

For the achieved expected latencies, $L_{kd}(\tilde{\mathbf{B}}_t)$, as opposed to those estimated at the nodes, the proof is more straightforward. For a given decision matrix, $\tilde{\mathbf{B}}_t$, the expected latencies $L_{kd}(\tilde{\mathbf{B}}_t)$ s are functions of the true meeting rates λ and are thus not random variables. Thus, the sequence $\{L_{id}(\tilde{\mathbf{B}}_t)\}$ is only a function of the random sequences $\{\hat{\lambda}_{ij}\}$, $j \in \mathcal{S}_i$ via the optimization that determines $\tilde{\mathbf{B}}_t$. Since we have established that $\tilde{\mathbf{B}}_t$ converges in probability to \mathbf{B}^* , it follows that $\{L_{id}(\tilde{\mathbf{B}}_t)\}$ converges in probability to $L_{id}(\mathbf{B}^*)$ due to property 1.

□

REFERENCES

- [1] A. Voyiatzis, "A survey of delay- and disruption-tolerant networking applications," *J. Internet Engineering*, vol. 5, no. 1, 2012.
- [2] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," in *IEEE Commun. Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.
- [3] Y. Cao and Z. Sun, "Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges," in *IEEE Commun. Surveys & Tutorials*, vol. 15, no. 2, pp. 654–677, 2013.
- [4] K. Wei, X. Liang, and K. Xu, "A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues," in *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 1, pp. 556–578, 2014.
- [5] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *ACM Sigplan Notices*, vol. 37, no. 10, 2002, pp. 96–107.
- [6] T. Small and Z. J. Haas, "The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proc. ACM Int. Symp. Mobile Ad hoc Net. & Comput. (MobiHoc)*, Annapolis, MD, Jun 2003, pp. 233–244.
- [7] B. E. Pataki and L. Kovács, "Sensor data collection experiments with chaoster in the fed4fire federated testbeds," in *IEEE Wireless and Mobile Comput., Net. and Comm. (WiMob)*, Larnaca, Cyprus, Oct. 2014.
- [8] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *IEEE Computer*, vol. 37, no. 1, pp. 78–83, Jan. 2004.
- [9] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 821–834, May 2012.

- [10] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke Univ., Durham, NC, USA, Tech. Rep., 2000.
- [11] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proc. IEEE Infocom*, vol. 3, Anchorage, AL, USA, Apr. 2001, pp. 1360–1369.
- [12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop on Delay-tolerant Networks*, Philadelphia, PA, USA, Aug. 2005, pp. 252–259.
- [13] J. A. Davis, A. H. Fagg, and B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," in *Proc. IEEE Int. Symp. on Wearable Computers*, Zurich, Germany, Oct. 2001, pp. 141–148.
- [14] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Comput. and Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2003.
- [15] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE Infocom*, vol. 6, Barcelona, Spain, Apr. 2006, pp. 1–11.
- [16] E. P. Jones, L. Li, J. K. Schmidtke, and P. A. Ward, "Practical routing in delay-tolerant networks," *IEEE Trans. Mobile Comp.*, vol. 6, no. 8, pp. 943–959, 2007.
- [17] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proc. ACM Int. Symp. Mobile Ad hoc Net. Comput. (MobiHoc)*, Montreal, Canada, Sep. 2007, pp. 32–40.
- [18] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, 2011.
- [19] D. A. Sharma and M. Coates, "Contact graph based routing in opportunistic networks," in *Proc. IEEE Global Conf. Sig. and Info. Proc. (GlobalSIP)*. Austin, TX, USA: IEEE, Dec. 2013.
- [20] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *IEEE Trans. Computers*, vol. 63, no. 7, pp. 1682–1695, 2013.
- [21] Y. Li, G. Su, D. O. Wu, D. Jin, L. Su, and L. Zeng, "The impact of node selfishness on multicasting in delay tolerant networks," *IEEE Trans. Veh. Tech.*, vol. 60, no. 5, pp. 2224–2238, 2011.
- [22] P. Sermpezis and T. Spyropoulos, "Understanding the effects of social selfishness on the performance of heterogeneous opportunistic networks," *Computer Commun.*, vol. 48, pp. 71–83, 2014.
- [23] H. Zhang, Z. Zhang, and H. Dai, "Gossip-based information spreading in mobile networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5918–5928, 2013.
- [24] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE J. Sel. Areas in Commun.*, vol. 26, no. 5, pp. 773–782, 2008.
- [25] M. Xiao, J. Wu, C. Liu, and L. Huang, "Tour: Time-sensitive opportunistic utility-based routing in delay tolerant networks," in *Proc. IEEE Infocom*, Turin, Italy, Apr. 2013, pp. 2085–2091.
- [26] C. Boldrini, M. Conti, and A. Passarella, "Modelling social-aware forwarding in opportunistic networks," in *Proc. PERFORM (LNCS 6821)*. Vienna, Austria: Springer-Verlag, Oct. 2010, pp. 141–152.
- [27] —, "Performance modelling of opportunistic forwarding with imprecise knowledge," in *Proc. Int. Symp. Model. and Opt. in Mobile, Ad Hoc and Wireless Net. (WiOpt)*, 2012, pp. 216–223.
- [28] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proc. ACM Int. Workshop on Mobile Opportunistic Networks*, San Juan, Puerto Rico, Jun. 2007, pp. 62–66.
- [29] M. Khouzani, S. Eshghi, S. Sarkar, N. B. Shroff, and S. S. Venkatesh, "Optimal energy-aware epidemic routing in DTNs," in *Proc. ACM Int. Symp. on Mobile Ad hoc Net. & Comput. (MobiHoc)*, South Carolina, USA, Jun. 2012, pp. 175–182.

- [30] D. J. Klein, J. Hespanha, and U. Madhow, "A reaction-diffusion model for epidemic routing in sparsely connected MANETs," in *Proc. IEEE Infocom*, San Diego, USA, Mar. 2010, pp. 1–9.
- [31] Q. Wang and Z. J. Haas, "Analytical model of epidemic routing for delay-tolerant networks," in *Proc. ACM Int. Workshop on High Perf. Mobile Opportunistic Sys.*, Paphos, Cyprus, Oct. 2012, pp. 1–8.
- [32] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a dtn routing protocol-prophetv2," in *Proc. ACM Workshop on Challenged Networks*, Las Vegas, NV, USA, Sept. 2011, pp. 27–30.
- [33] S. Shaghaghian and M. Coates, "Opportunistic networks: Minimizing expected latency," in *IEEE Wireless and Mobile Comput., Net. and Comm. (WiMob)*, Larnaca, Cyprus, Oct. 2014.
- [34] H. Cai and D. Y. Eun, "Crossing over the bounded domain: from exponential to power-law intermeeting time in mobile ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1578–1591, Oct. 2009.
- [35] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Trans. Mob. Comput.*, vol. 6, no. 6, pp. 606–620, Jun. 2007.
- [36] V. Conan, J. Leguay, and T. Friedman, "The heterogeneity of inter-contact time distributions: its importance for routing in delay tolerant networks," 2007, arXiv:cs/0609068v2 [cs.NI], LIP6.
- [37] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in *Proc. ACM MobiHoc*, New Orleans, LA, USA, May 2009, pp. 299–308.
- [38] K. Lee, Y. Y. amd J. Jeong, H. Won, I. Rhee, and S. Chong, "Max-contribution: On optimal resource allocation in delay tolerant networks," in *Proc. IEEE Infocom*, San Diego, CA, USA, Mar. 2010, pp. 1–9.
- [39] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. M. Ni, "Recognizing exponential inter-contact time in VANETs," in *Proc. IEEE Infocom*, San Diego, CA, USA, Mar. 2010, pp. 1–5.
- [40] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2006-01-31)," Downloaded from <http://crawdad.org/cambridge/haggle/>, Jan. 2006.
- [41] R. H. Berk, "Consistency and asymptotic normality of MLE's for exponential models," *The Annals of Math. Stat.*, vol. 43, no. 1, pp. 193–204, 1972.